

Privacy in Enterprise Identity Federation

- Policies for Liberty 2 Single Signon -

Birgit Pfitzmann¹
IBM Zurich Research Lab
bpf@zurich.ibm.com

Abstract. Cross-domain identity management is gaining significant interest in industry. A well-known example is the Liberty Alliance's specifications for single signon of web users across different enterprises. The Liberty Alliance stresses that account linking is voluntary for the users and that privacy is an important consideration. We evaluate the privacy of these specifications in detail. We point out some ambiguities and propose a concrete privacy policy together with a few changes to the Liberty processing rules. Our analysis demonstrates that identity-management policies need detailed advance planning even in a limited context.

Keywords. Identity management, federation, single-signon, privacy, pseudonyms, user tracking.

1 Introduction

Identity management has many facets. In enterprises, the focus is still on internal consolidation, e.g., on customer-relationship management and on integrating different access channels. In the privacy-research community, the focus is on enabling people to manage their identities themselves including free choice of roles and pseudonyms, the transfer of credentials from one pseudonym to another pseudonym of the same person, and appropriate user interfaces. The gap between these facets is wide. Cross-domain web single-signon proposals like the Liberty Alliance's are somewhere in the middle: They are not intended to solve the main legacy identity-management problems in an enterprise. Nevertheless the main commercial interest in them is for single signon between loosely coupled parts of one enterprise, and for small groups of enterprises with pre-existing contracts, e.g., supply chains or airlines mileage programs. However, the last example shows that commercial interest extends to the consumer market, where privacy is a serious issue. The first cross-domain web single-signon proposal, Microsoft Passport, was even entirely geared towards the consumer market.

For the consumer market, surveys show that lack of user trust is a major inhibiting factor for electronic commerce, and that a large majority of the population is concerned about privacy. About 25% are willing to pay a considerable price for it in money or inconvenience; most others want enterprises to build privacy in voluntarily or the state to force enterprises to do so. This means that a large group of consumers are sufficiently worried not to use a lot of enterprise-side services, while not sufficiently motivated to buy and set up user-side privacy technology. For business-to-business scenarios, concerns about the privacy and commercial value of customer data and the fact that employee data and actions may involve business secrets limit flexible cross-domain business scenarios. Both concerns can be handled by privacy techniques.

¹ This paper reflects the view of the author, which is not necessarily shared by IBM. IBM is not a member of the Liberty Alliance.

A preliminary version of this paper was presented at the 3rd Workshop on Privacy Enhancing Technologies (PET 2003), LNCS, Springer-Verlag, Berlin Heidelberg 2003

Specifically for web single signon and federated identities, market studies in the wake of Microsoft's Passport product [Mic01] corroborated clearly that lack of privacy and choice is a large concern and limits adoption. The Liberty Alliance therefore stressed from its beginning that federating, i.e., choosing single signon between two enterprises, is voluntary for the users and that privacy will be protected. The Liberty specifications [Lib02,Lib03] contain concrete provisions for privacy, such as pseudonyms. In this paper, we investigate the achieved privacy for the central single-signon part of these specifications and make additional proposals. We have chosen the Liberty specifications for this analysis primarily because they currently contain the most explicitly fixed protocol and privacy details. Further, the strong membership in the Liberty Alliance make them an important proposal. Microsoft Passport also has explicit policies, but a globally fixed person identifier is sent to every party who wants to track a user. This is so privacy-unfriendly that further considerations are moot. The OASIS standard SAML (Security Assertion Markup Language) [SAM02] is very generic, i.e., a security section states that privacy must be considered, but concrete decisions are left to the applications. Similarly, the broader WS-Federation Passive Client proposal strongly stresses privacy but is policy neutral [WSF03]. For an abstract discussion of privacy requirements and design options for web single signon see [PW02].

Actually, the balance between concreteness and generality has also shifted from Liberty Phase 1 (with single-signon versions 1.0/1.1) to Phase 2 (with single-signon version 1.2). Phase 1 focussed on small groups of enterprises with close trust relationships, called circles of trust; the protocols did not scale to large and multiple groups. This led to a fixed protocol with few options and relatively clear privacy implications. Phase 2 allows far more choices already in the single-signon protocols although the exchange of real-world user attributes is still mainly outside these protocols.

The minimum goal of a deployment of a single-signon standard with respect to privacy should be clarity: If users believe in stronger privacy than enterprises do, the users will feel cheated and may start litigation. If enterprises believe in stronger privacy than users do, users will be more reluctant to use the protocols than they need to be. For a standard with such a narrow functionality as single signon, we believe this clarity is best approached in the standard itself, leaving only a small number of easy-to-understand options for users and organizations. Studying a standard under this point of view has two goals: First, make sure that the privacy policies and implications are clearly specified. This is a completely technical goal. Secondly, discuss whether these policies, including the given user options, are suitable for the stated purposes. A third potential goal is out of scope of our paper: to compare such policies with exact privacy regulations for different countries and sectors. However, we hope that our technical work can serve as a basis for such legal studies.

2 Liberty Single Signon and Federation

We first introduce web single-signon protocols in general, and then special aspects of the Liberty protocols.

2.1 Single Signon with Browsers

The Liberty specifications are one of several recent specifications of cross-domain web single signon across different enterprises for users that have nothing but a browser [Mic01, SAM02, Lib02, Lib03, WSF03]. Including this *browser-only* or *zero-footprint* case is currently

considered essential for market acceptance. It should comprise users who frequently switch browsers or share browsers with others. Good protocols can work without any active content or cookies, while most take advantage of JavaScript and cookies when a browser enables them.

The overall structure of all these protocols is shown in Figure 1. A user is initially browsing at a service provider. When the user wants to log in, the service provider redirects the browser to an identity provider of the user. The user logs in there, typically with a fixed user ID and password. The browser and identity provider may also reuse a secure session from another recent login. The identity provider then redirects the browser back to the service provider with some ticket. If the information to be transferred is short, it can be completely included in this ticket. Most protocols also provide a back channel for transferring longer information; the ticket then contains a handle to that information so that the service provider can associate a returning browser with the appropriate back-channel information.

We speak of *single signon* if only a login name or pseudonym is transferred, and of *attribute exchange* if other user information like address or credit card number is transferred.

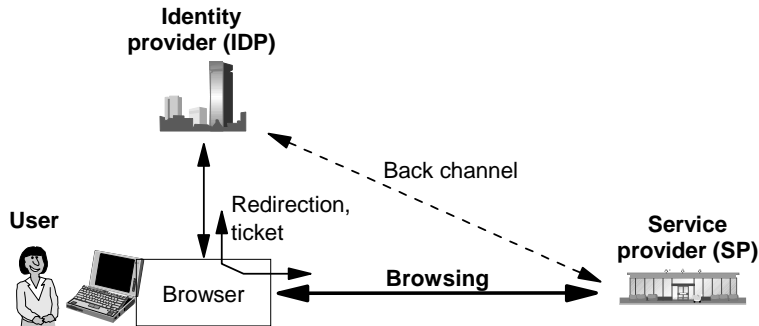


Figure 1 Scenario of browser-based single signon

2.2 Special Aspects of Liberty Single Signon

The overall Liberty Phase 2 single-signon scenario is described in [Lib03a]; for a general overview see [Lib02a]. A user starts participating by consenting to *introductions* at an identity provider. Such an initial phase is normally called registration; but in Liberty it is assumed that the user already has an account at the identity provider. Later, when the user browses at a service provider, the service provider notices that the user has an identity provider, and asks the user whether he wants to *federate*, in other words link, these two accounts. The only subprotocol specified for how the service provider notices this is by a cookie in a common federation domain; we assume in the sequel that this subprotocol is used. Then the redirections as in Figure 1 happen for the first time, and the identity provider and service provider exchange a pseudonym by which they will refer to this user. Later signon happens under this pseudonym, again according to Figure 1. It is usually pure single signon without attribute exchange. The message formats are described in [Lib03b] and the interaction protocols, called profiles, in [Lib03c]. The structure in Liberty Phase 1 was similar, but the versions differ in several details that are important for privacy, as discussed below.

The informal parts of Liberty Phase 2 still suggest that introductions and linking only happen within *circles of trust*. Those were well-defined in Phase 1 by out-of-band metadata

exchanges. Phase 2, however, allows arbitrary trust models, including online exchange of certificates; hence a circle of trust may be highly dynamic or comprise the entire world.

In addition, Liberty Phase 2 contains web service standards for exchanging information on back channels. The pseudonyms from single signon can be used as person designators in these protocols. Further, both phases contain a single-signon profile for specific Liberty-enabled client programs. There are also non-normative security and privacy best practices [Lib03d]. However, they mainly advise on privacy laws and guidelines without many concrete relations to the specifications.

Our privacy analysis is only intended to be complete for the browser single-signon protocol of Liberty Phase 2. Analyzing all parts is beyond the scope of one paper. An analysis of Phase 1 can be found in our preliminary workshop paper.

2.3 Related Work

We already mentioned other important cross-domain web single-signon proposals above. Scientific contributions mainly investigated the operational security [KR00, Sle01] and man-in-the-middle security [PW03a, Gro03] of such proposals. We gave an overview of privacy requirements and design consequences for cross-domain web single-signon protocols in [PW02], together with a sketch of a protocol BBAE achieving optimal privacy. BBAE is described in more detail in [PW03].

The high end of privacy-enabled identity management is exemplified by the idemix prototype of an anonymous credential system [CV02]. Anonymous credentials were first proposed in [Cha85]. They are the only known solution for situations where a user wants to interact in unlinkable roles with different organizations and nevertheless transfer certified attributes between these organizations. However, in current electronic commerce, not many attributes are certified; typically users just fill in forms. Even Microsoft Passport did not certify anything until quite recently, and now only control of an email address. Hence one can strive for using simple pseudonyms where no certification is needed, simple certificates where no anonymity is possible for other reasons (which may, however, be non-technical and thus changeable), and anonymous credentials in the remaining case.

For a discussion of more remotely related techniques like form fillers and PKIs, we refer to the appendix of [PW02].

3 Single-Signon Data Analysis and Policy Proposals

We now approach the privacy question systematically. We identify all data exchanged in the Liberty single-signon protocols and all user consents mentioned in the specifications. We propose concrete policy rules for these data categories, using the available types of consent as far as possible. We propose to fix one clear policy for usage with Liberty single-signon or similar specifications, leaving a larger choice of policies to attribute-exchange protocols, which are separate in Liberty. In other words, simplicity is the main advantage of factoring out single signon, and this should be reflected by a clear and simple policy.

3.1 Liberty Consents

According to the user-experience figures in [Lib03a], a user has two choices regarding Liberty single signon, corresponding to opt-ins to certain data releases:

1. When logged in at an identity provider *IDP*, the user can allow so-called introductions to service providers. We call this *introduction consent*.
2. When interacting with a service provider *SP*, the user can allow to federate her current identity with that at an identity provider *IDP*. We call this *federation consent*. The use cases suggest that the user must be logged in at *SP*, and thus a priori known at *SP*, but technically nothing is based on this.

The normative parts of the specification are consistent with this, but less explicit. The only seemingly normative sentence involving consent is “If `<NameIDPolicy>` is *federated*, and if the Principal consents, then the identity provider MAY federate the Principal’s identity with the requesting provider ...,” but even this does not imply that federating is forbidden without consent. However, the formulations in [Lib03a] would be grossly misleading if these two types of consent were not required, and we will propose suitable rules in the following.

In addition, the specification contains message fields for transporting consent, but no rules for the recipient based on them. The rule for the sender is to indicate “whether or not consent has been obtained from a user in sending this message”. This leaves open whether there was explicit consent to all fields of the message, and whether the consent involves actions typically triggered by the message. Auditability of the consent is informally required but impossible to achieve in most cases for browser users.

The best practices often mention permissions-based attribute sharing besides single signon, and once permissions are mentioned in the context of identity providers and thus for single signon. Further, a policy is assumed in two normative sentences “If the `<NameIDPolicy>` element is *federated*, then a new identity federation MAY be created, if one does not already exist for the Principal and policy permits” and “If the policy for the Principal forbids federation, then evaluation MAY proceed as if the value were *onetime*.” Again this does not literally imply that federating is forbidden in other cases, but we propose to require the same consent as above.

3.2 Introduction Data

Upon introduction consent, *IDP* sets a cookie on the user’s browser in a federation domain. Recall that we assume that the only specified introduction protocol is used [Lib03c, Section 3.6]. This cookie tells all other members of this domain that the user has identity provider *IDP*. It is explicitly stated that the cookie may be either persistent or session-based. Thus it either divulges the name *IDP* or that the user has recently logged in at *IDP*. Hence it corresponds to the following lax privacy rule.

Rule_{intro}: If user *U* gives introduction consent, *IDP* may tell arbitrary parties where *U* is browsing that this user has identity provider *IDP*, and whether *U* has recently logged in at *IDP*. This holds until future opt-out by revoking introduction consent; an easy interface for this must be provided.

Our rule does not restrict the release to a circle of trust. Recall that those would be hard to define with the flexible trust models of Liberty Phase 2. Even in Phase 1 it would be problematic to restrict consent to a circle of trust: Before consenting, the user should be able to look up the members of this circle. However, then the consent would not extend to members added later, while new members automatically get the cookie. Figure 4 of [Lib03a] should be modified accordingly, or an implementation of a more restricted introduction should be proposed.

Although lax, this rule is not unreasonable, because the cookie data that introduce an identity provider are not very sensitive. They are not linked to a name of U , only “the current browser user” and contain no details like the responsibilities of a particular identity provider with respect to a user. (However, the corresponding element in the Liberty-enabled client profile, a header, is extensible. More detailed privacy rules are needed there.) Releasing these data improves user convenience, which seems the main motivation for a pure single-signon protocol. For users who withhold this broad introduction consent, service providers should query the user for an appropriate identity provider. This option is needed anyway for users who switch browsers or share browsers with others, i.e., for the real zero-footprint case.

In addition, as a common domain for introduction cookies is a powerful tool that can be misused by many other cookies, we propose the following rule:

Rule_{cookies}: The common domain is not to be used for cookies except the specified introduction cookies.

We propose that introduction consent has no effect for a service provider SP ; in particular SP should not collect introduction data or contact IDP unless a user chooses to federate at SP .

3.3 Main Single-Signon Data: Pseudonyms

As mentioned in Section 2.2, Liberty has a notion of “federating” or account linking, which should happen before single signon. Technically, this is one option of the single-signon protocol shown in Figure 1. Hence we immediately analyze the single-signon protocol. The primary data released by the identity provider IDP in single signon is a pseudonym. In this section, we consider the privacy implications of this pseudonym release. Liberty Phase 1 only defined long-term pseudonyms; Phase 2 has added one-time pseudonyms. For both types, the specifications strongly suggest that different pseudonyms are unlinkable, i.e., it is impossible to see whether two pseudonyms refer to the same user. We will assume this, but it could be formulated more clearly.

3.3.1 Long-Term Pseudonyms

The main privacy impact of a long-term pseudonym is its repeated release, which allows user tracking across different transactions. Hence what federation consent is about is the permission to track the user. This is particularly critical if a service provider already knows an identity of the user at the time of federation, as suggested by [Lib03a].

Liberty only allows a user one long-term pseudonym, i.e., one role, per service provider, given one account with an identity provider. This is a limitation. It resembles the pseudonym policies of [GGK+99].

Liberty Phase 1 restricted the user-tracking abilities by making each long-term pseudonym specific to one service provider. Phase 2 has relaxed this by introducing affiliations. By forming an affiliation, several service providers can share user pseudonyms and thus enable joint user tracking. A statement that affiliations must be a-priori known to the identity provider somewhat limits abuse of this concept, but there is no clear rule or relation to consent. We propose that federation consent is needed for affiliations as well as for service providers.

We also propose that, in contrast to the user-interface examples in [Lib03a], the federation consent allowing the establishment of a long-term pseudonym must be given at IDP . We recommend the following rule for identity providers:

Rule_{IDP,auth}: User U can separately give federation consent at IDP for every specific service provider SP or affiliation A to allow federation with SP or A . For an affiliation, the consent form must contain a link to the current affiliation membership and to clear rules whether members may be added later. Then IDP may authenticate U under a fresh, but then fixed role pseudonym $id_{U,SP}$ to SP whenever SP asks, respectively under a pseudonym $id_{U,A}$ to members of A whenever those ask. This holds until future opt-out by federation termination.

The processing rules in [Lib03b, Section 3.2.2.6] should make the need for this consent explicit. It may, however, occur earlier within consent to a policy. Federation consent is independent of introduction consent in this rule.

The advantages of this rule over giving federation consent at SP are enormous. First, some laws require direct consent and notification. Secondly, IDP would otherwise have to believe SP whether it got consent. Identity providers can certainly not trust all service providers to whom they can provide user authentication; hence a special trust infrastructure would be needed and provisions for federation consent at IDP for the untrusted SP s. Thirdly, the only way to audit claimed consents would be that IDP gives the users access to the list of federated SP s (because federation may happen without user interaction, in particular if SP sends a request with the element `<isPassive> = true`), and disputes would be almost impossible to resolve.

Federation consent at a service provider SP is still needed for the following proposed rule.

Rule_{SP,auth}: If user U gives federation consent for an identity provider IDP at SP , then SP may record this choice (e.g., by setting a cookie) and the obtained pseudonym $id_{U,SP}$ of this user, and may link different interactions with this user by this pseudonym. If the choice was made from an existing account, it may also link these interactions to the existing account.

3.3.2 One-Time Pseudonyms

A one-time pseudonym is used only once. No user consent seems to be required for single signon under a one-time pseudonym p in Liberty, and at first glance it seems to have no privacy impact. However, if the user is already logged in at the service provider SP when such a single signon takes place, then p enables IDP and SP to link the two accounts, which might otherwise be impossible or require complicated data matching. This also holds for anyone later getting hold of both records. Further, p is almost certainly used for a subsequent attribute exchange. We propose the following privacy rule:

Rule_{IDP,onetime}: If user U has given introduction consent at IDP , then IDP may authenticate U under unlinkable one-time pseudonyms to arbitrary service providers. IDP has to delete these pseudonyms and transaction records within 1 day, unless there is special user consent for longer retention, or an attribute release based on p follows where a degree of certainty is promised that requires longer retention (such as dispute resolution) and the user has consented to that. Without introduction consent, IDP needs U 's consent for each authentication.

3.4 Other Single-Signon Data Releases by Identity Provider

So far we concentrated on the pseudonym that the identity provider releases to a service provider during single signon, because this is the one explicitly desired personal information

element. Now we study whether other message parts carry personal information. Indeed this is possible in the Liberty protocols.

3.4.1 Direct Personal Information

We first propose the following rule:

Rule_{IDP,explicit}: An identity provider must not put personal information of a specific user into any field of any message of a single-signon protocol except within a `<Statement>` in an `<Assertion>` in the `<AuthnResponse>`.

This seems to be intended, but should be made a mandatory processing rule. Examples of Liberty message parts that might in principle destroy privacy are the `<Extension>` element of `<AuthnResponse>`, the `<Advice>` in an assertion and additional fields in the query string of a redirect with SAML artifact.

Now the overview in [Lib03a] suggest that only authentication statements are contained, e.g., the names `<AuthnResponse>` and “authentication assertions” and the text after Figure 7 stating “cleartext identifiers WILL NOT be exchanged”. Then the pseudonym is indeed the only transferred personal information. However, [Lib03b] explicitly allows arbitrary SAML statements, and thus in particular attribute statements. It seems unreasonable to require that attribute statements do not contain personal information. Indeed, attribute statements can often be a much simpler way of transferring attributes, with possible online consent or corrections by the user, than an additional web-service channel and the need to use a so-called interaction service as proposed in Liberty. We require that user attributes in a `<Statement>` are handled according to the Rule `RuleIDP,attributes` that we propose below.

3.4.2 Provider Information as Personal Information

The formulation “of a specific user” in `RuleIDP,explicit` was chosen to tolerate that several message fields contain information about the identity provider, which is indirect information about the user. E.g., a small company as identity provider restricts the possible users to very few employees. This was one reason why we bound even the release of one-time pseudonyms to at least introduction consent in Section 3.3.2. For long-term pseudonyms, our rules bind this information release to federation consent.

Liberty single signon has one real limitation in this respect: It does not provide for local wallets, i.e., the option that some users store some attributes on their own machines while remaining anonymous. Local wallets are perfectly reasonable for most attributes because current electronic commerce does not rely on external certification of most user information, and for attributes like preferences the user is even the primary identity provider. Of course, users of local wallets have to give up the zero-footprint option. In Phase 1, with small circles of trust and without attribute exchange, this was no problem, but in the wide scenario of Phase 2 local wallets would be an important option. The problem is that identity provider information is strongly constrained in Liberty. An anonymous local wallet needs a relative address and one name (key) per user role. (Details can be found in [PW03].) It may be possible to tweak the constraints because most of them only concern metadata, but prescriptions like URI-based identifiers, certificate paths, established relationships, and resolvability of a Provider ID to a URL in [Lib03b] would better come with special provisions for local wallets.

3.5 Traffic Data from Service Providers

Traffic data arise at an identity provider *IDP* during single signon because the service provider *SP* notifies *IDP* that the user is browsing there. More precisely, first *IDP* learns that the user whom *IDP* knows as *U* is currently browsing at *SP*, while *SP* itself does not know this. This allows *IDP* to track user behavior. As these data arise implicitly in the protocol and are not needed for the applications that a user expects, we propose strict privacy rules for them. We start with the identity provider.

Rule_{IDP,traffic}: An identity provider *IDP* must not mine traffic data, use them for any other purpose than single signon, or forward them to another party. If a single signon fails due to lack of consent, the corresponding traffic data must be deleted immediately unless the user requested to be notified of such events. Exceptions may only be given by law, e.g., storage requirements for law enforcement when single signon is provided, and for authentication classes where dispute resolution is offered.

For the service providers, the Liberty user-experience overview suggests that single signon only occurs after federation consent ([Lib03a], Sections 2.2 and 5.4.2). However, Phase 2 allows signon under one-time pseudonyms without federation consent. Further, when *SP* desires single signon, it often does not know whether it has federated for the current user *U* because *U* is unidentified at this moment. We therefore introduce a new type of consent, single-signon consent, that the user can give to a service provider for one single signon, and propose the following rule:

Rules_{SP,traffic}: A service provider *SP* needs federation consent or single-signon consent for an identity provider *IDP* from a user before making a single-signon request for this user to *IDP*. Given the consent, it must only provide fixed data about itself to *IDP* in the single-signon protocol, unless there is separate consent for user-dependent data.

An alternative would be to allow single-signon requests also if *SP* sees an introduction from an identity provider. This would have several disadvantages: First, in some laws service providers need direct user consent for data releases. Secondly, the user-interface example for introduction consent from [Lib03a] would become misleading, and describing the full implications may become so long that few people will consent. Thirdly, one would need strict audit to prevent dishonest parties from introducing themselves as identity providers for unsuspecting users to collect visited-sites trails from service providers.

Situations where Rule_{SP,traffic} requires user interaction typically require user interaction anyway, so that no convenience is lost: If the user has no cookies (switched off or changed browser), then user interaction is needed to determine a suitable identity provider. If the user has cookies, but not from *SP* yet, then typically *SP* will ask for federation consent or user interaction will be needed during attribute exchange under a one-time pseudonym.

A consequence of this rule is that *SP* should not put unencrypted user data in the element `<RelayState>` of a single-signon request, e.g., the exact URL that the user wanted to access. This fits [Lib03b, Section 3.2.1.1], while [Lib03c] still contains a cleartext URL in examples, corresponding to a different recommendation in Phase 1.

An additional feature of Liberty Phase 2 that produces traffic data is proxying. This means that an identity provider *IDP* may ask a second identity provider *IDP'* for the actual user authentication. The service provider can prevent proxying, but no user consent or policy is mentioned in this context. The proxying request is a normal single-signon request by *IDP* without mentioning *SP*. Hence it conforms to the second sentence of Rule_{SP,traffic}, with *IDP* in

the role of service provider, and we simply propose that this rule also applies for proxying. For *IDP*, all identity-provider rules apply anyway because it does not notice the proxying.

4 Privacy Beyond Single-Signon Data

So far, we have analyzed the data released in Liberty introductions and single signon and proposed policies governing these releases. In this section, we consider related protocols and further aspects of privacy policies.

4.1 Attribute Exchange

According to [Lib03d], the Liberty Phase 2 specifications are intended to “support and promote permissions-based attribute sharing to enable a user’s (“Principal’s”) choice and control over the use and disclosure of such Principal’s personal information”. While analyzing the web services defined for this purpose is outside the scope of this paper, we have to analyze the privacy impact of single signon as an enabler of attribute sharing. This impact is considerable: The common name (pseudonym) for a user that an identity provider *IDP* and a service provider *SP* have after single signon enables attribute exchange in many situation where it would be impossible or hard otherwise.

Two principles seem clear, although the second one is not explicit in the Liberty specifications:

- In contrast to single signon, we cannot propose one fixed policy (even with user options) for all enterprises. Nevertheless, we would specify certain minimum requirements on policies, such as dispute resolution and protection standards, while the Liberty Alliance did not.
- The introduction, federation and single-signon consent discussed so far do not imply consent to the exchange of further attributes.

We propose, however, that federation or single-signon consent allow a service provider *SP* to *attempt* to obtain further information about the user from *IDP*, i.e., to send attribute requests under the obtained pseudonym to *IDP*. Such requests have privacy impact because the selection of requested attributes may depend on user actions. However, it would seem confusing to ask users for special consent for this. For more sensitive data or under certain laws, the service provider may need direct user consent for *receiving* the attributes; that consent should be asked before making the request. We propose the following rules:

Rule_{IDP,attributes}: If user *U* consents at *IDP* to federate with a service provider *SP*, then *IDP* may use the generated pseudonym $id_{U,SP}$ to provide information about *U* to *SP*, provided *U* also consented (during the transaction or earlier via a privacy policy) to the sending of this particular information to *SP*. If such a policy exists prior to federation consent at *IDP*, it must be explicitly referred to and easy to look up before this consent and at any later time, and withholding federation consent must be easy. The permission holds until future opt-out by defederation. To what extent sent attributes may survive defederation must be clarified in the policy, as well as to what extent *IDP* may store a history of *SP*’s requests.

Rule_{SP,attributes}: If user *U* consents at *SP* to federation or single signon with an identity provider *IDP*, then *SP* may use the obtained pseudonym $id_{U,SP}$ to ask *IDP* for reasonable attributes about *U*. It may use the obtained attributes in its current transaction, and it may further use or store them according to privacy policies consented to by *U*, where it may

trust non-repudiable statements by *IDP* about *U*'s consent unless regulations forbid this. If the consent is given at *SP*, the policy must be explicitly referred to and easy to look up before the consent, and withholding consent must be easy. A permission based on federation consent holds until future opt-out by defederation, one based on single-signon consent for the duration of the transaction. To what extent received attributes survive defederation or the transaction, respectively, must be clarified in the policy.

The requirement to show a prior attribute-exchange policy to the user at federation consent is a balance between user needs for clarity and enterprise legacy issues: Our formulation only refers to attribute-exchange processes using $id_{U,SP}$. Hence it does not restrict legacy processes. However, all processes using $id_{U,SP}$ are necessarily new and known. Therefore one can set up an explicit privacy policy for them, even when this policy is based on prior indirect consent, e.g., of employees to employers to share data with certain business partners. The overview [Lib03a] would be misleading if this requirement were omitted, because it contains several statements like that federation consent alone “WILL NOT” lead to attribute exchange.

Further note that the proposed rules are asymmetric, i.e., attributes are only transferred from *IDP* to *SP*. This is more user-friendly given the distinction between identity providers and service providers. The permission for *SP* to trust non-repudiable statements by *IDP* corresponds to the sticky-policy paradigm [KSW02]; the mechanism in Liberty web services is usage directives. As a default, i.e., without a directive, we proposed a very strict rule for *SP*'s use of the attributes.

4.2 Personal Data on Lower Layers

Lower-layer protocols triggered by single signon may also convey personal data. Typical examples are traffic analysis in networks, data released automatically by browsers, and interactions with central directories like key-distribution centers. We are not aware of specific effects of Liberty single signon on the first two possibilities. Communication with directories should not involve any user data (including traffic data) unless separate consent is given. Except in the case of local wallets this should be easy, and local wallets, as we saw above, are unfortunately not easily possible in Liberty protocols at present.

4.3 Further Policy Aspects

So far, we have considered rules for the data releases from one party to another. Privacy policies also govern other aspects of data handling. An overview of such aspects in different regulations and guidelines is given in [Lib03d]. While we also recommend that these aspects should be fixed as far as possible for a single-signon standard, we only sketch specific aspects in the context of single signon.

- **Termination:** There is no Liberty protocol for revoking introduction consent. We propose to add it and to require that the identity provider deletes its name in the introduction cookie. This is also useful if a user changes identity providers. Single-signon consent is per transaction and does not need revocation. Federation consent can be revoked at either *IDP* or *SP* by “federation termination” [Lib03c]. Either party has to notify the other. If *SP* and *IDP* later federate again, *IDP* generates a new pseudonym. Hence a user can reliably cut the link between its prior interactions with *SP* and future ones, unless federation occurs both times from an existing account with *SP*.

- **Dispute resolution:** A minimum standard for dispute resolution should be set. We recommend at least a contact address at each *IDP*, one for each defined circle of trust, and one for each conformance program as a last resort. Further, while law suits are then hopefully not needed, a breach of privacy promises can be a basis for litigation even if this is not offered as a dispute resolution type in the policy.
- **Notification:** Under the recommended policy, no minimum standard for user notification is needed, because all data releases are governed by policies that were consented to at first such release.
- **Access rights for the user:** Under the recommended policy, no user access rights are critical except for attribute usage by a service providers based on a policy statement by *IDP*. Generally, policies where one party believes a second party that a user gave consent at the second party need access rights to enable misuse detection. Access to the names of service providers federated with an identity provider *IDP* is useful and also needed in the user interface for defederation at *IDP*.
- **Retention periods:** Each identity and service provider should state retention periods for all data covered by these policies. In addition, a general upper bound on the retention of traffic data seems useful, e.g., one month unless local law or the authentication class require a longer period.
- **Assurance:** Certain minimum assurance standards should be fixed at least for identity providers, e.g., that the policy has a recognized privacy seal. Attribute policies at identity providers can additionally require assurance for service providers that receive certain attributes.

5 General Recommendations

The policy that we recommended for the Liberty 2 single signon protocols is summarized in Table 1.

Data category	Detailed data	Consent needed at	Exists in Liberty SSO?	Recommendation for Liberty SSO policy	Recommended side effects of rule
General					Later termination; minimum dispute resolution and protection standards. Seal for policy.
Introduction	<i>U</i> 's identity provider	<i>IDP</i>	Yes	Do after introduction consent	Retention limit
	Login status	<i>IDP</i>	Yes	Do after introduction consent	n/a
Traffic	Name of <i>SP</i> and fact that <i>U</i> is now browsing there	<i>SP</i>	Yes	Do after federation or single-signon consent at <i>SP</i>	Retention limit
	Anything else about <i>U</i> 's current actions at <i>SP</i>	<i>SP</i>	Maybe	Don't	n/a
Names	Fixed role name per <i>SP</i>	<i>IDP</i>	Yes	Do after federation consent at <i>IDP</i>	Retention limit
Other user attributes	Arbitrary	<i>IDP</i>	Maybe	Consent to policy with federation consent at <i>IDP</i>	Details in policy
	What attributes <i>SP</i> wants	<i>SP</i>	Maybe	Consent to policy with federation consent at <i>SP</i>	Policy mainly covers <i>SP</i> 's usage of received attributes

Table 1 Summary of recommended policy rules for Liberty single signon

General limitations of the Liberty single-signon specifications, in particular for consumer shopping scenarios, are a lack of support for local wallets and the fact that user roles are essentially bound one-to-one to service providers. Recall further that combinations of anonymity and certification, although rare in current e-commerce, cannot be handled by zero-footprint solutions, only by cryptographic credentials.

While the Liberty specifications are the most explicit ones available not only with respect to privacy policies, but also with respect to user experience, we believe that the importance and difficulties of the user interface are still underestimated. Many figures in [Lib03a] do not fully reflect the consent that the user is giving at that moment. They also lack elements for withholding consent, aborting transactions, asking for help, and looking up privacy policies. We recommend that figures should always convey best privacy practices and help individual enterprises in designing correct and complete screens. Even standardization should be considered.

Privacy languages for expressing attribute-exchange policies are available [APP02, EPA03], but the challenge will be to guide users, including administrators, in setting appropriate policies. For the consumer shopping scenario, we believe that most data releases will, for a long time to come, have to be approved online during the transaction. Such real-time release is possible in principle with the Liberty interaction services and by local user interaction if attribute statements are included in single signon, but we believe that this feature deserves more attention.

6 Summary

We have analyzed the privacy impacts of a cross-domain web single-signon proposal, the browser single-signon protocols from the Liberty Phase 2 specifications. We did *not* analyze other privacy-sensitive Liberty specifications like the name identifier mapping and the web services. Although single signon is a very fixed concept compared with general attribute

exchange, we identified several privacy ambiguities. We proposed a precise policy and described small changes to Liberty's processing rules needed to support this policy, in particular two new types of consent. We also pointed out general aspects that might be added to specifications like Liberty's to enable fully user-controlled identity management, such as local wallets and freely chosen roles.

An underlying belief in our analysis is that if it is worth while factoring out and standardizing a rather fixed service like single signon on the messaging level, then it is also worth while standardizing a corresponding privacy policy. This increases clarity and user convenience, in particular as most users of cross-domain single signon will deal with multiple implementations of the same standard at different identity and service providers. Enterprises will benefit from the resulting wider adoption by users and faster transactions. We appreciate that the Liberty Alliance has taken steps in this direction, and hope that our analysis can aid the further development of these and other specifications.

Acknowledgements

Thanks to Susan Landau, Matthias Schunter and Michael Waidner for interesting comments on these policy recommendations.

References

- APP02 A P3P Preference Exchange Language 1.0 (APPEL1.0); W3C Working Draft, 15 April 2002, <http://www.w3.org/TR/P3P-preferences/>
- Cha85 David Chaum: Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM 28/10 (1985) 1030-1044
- CV02 Jan Camenisch, Els Van Herreweghen: Design and Implementation of the Idemix Anonymous Credential System; 9th ACM Conference on Computer and Communications Security (CCS), 2002, 21-30
- EPA03 IBM: Enterprise Privacy Authorization Language (EPAL); Submission request to W3C, <http://www.w3.org/Submission/2003/07/>, November 2003
- GGK+99 Eran Gabber, Phillip B. Gibbons, David M. Kristol, Yossi Matias, Alain Mayer: Consistent, Yet Anonymous, Web Access with LPWA; Communications of the ACM 42/2 (1999) 42-47
- Gro03 Thomas Groß: Security Analysis of the SAML Single Sign-on Browser/Artifact Profile; 19th Annual Computer Security Applications Conference (ACSAC 2003), IEEE Computer Society Press, 2003
- KR00 David P. Kormann, Aviel D. Rubin: Risks of the Passport Single Signon Protocol; Computer Networks 33 (2000) 51-58
- KSW02 Günter Karjoth, Matthias Schunter, Michael Waidner: The Platform for Enterprise Privacy Practices - Privacy-enabled Management of Customer Data; 2nd Workshop on Privacy Enhancing Technologies (PET 2002), LNCS 2482, Springer-Verlag, Berlin 2003, 69-84
- Lib02 Liberty Alliance Project: Liberty Phase 1 Specifications (6 parts); first published July 2002, final version http://www.projectliberty.org/specs/archive/v1_1/liberty-specifications-v1.1.zip
- Lib02a The Liberty Alliance Technology Expert Group: The Development of Open, Federated Specifications for Network Identity; Information Security Technical Report (ISTR) 7/3 (2002) 55-64
- Lib03 Liberty Alliance Project: Liberty Phase 2 Final Specifications, November 2003, <http://www.projectliberty.org/specs/lap-phase2-final.zip>
- Lib03a Liberty ID-FF Architecture Overview; in [Lib03]
- Lib03b Liberty ID-FF Protocols and Schema Specification, Version 1.2; in [Lib03]
- Lib03c Liberty ID-FF Bindings and Profiles Specification, Version 1.2; in [Lib03]
- Lib03d Liberty Alliance Project: Privacy and Security Best Practices, Version 2.0, Nov. 12, 2003, http://www.projectliberty.org/specs/final_privacy_security_best_practices.pdf
- Mic01 Microsoft Corporation: Various .NET Passport documentation (started 1999), in particular Technical Overview, Sept. 2001, and SDK 2.1 Documentation; <http://www.passport.com> and <http://msdn.microsoft.com/downloads>

- PW02 Birgit Pfitzmann, Michael Waidner: Privacy in Browser-Based Attribute Exchange; ACM Workshop on Privacy in the Electronic Society (WPES) 2002, ACM Press 2003, 52-62
- PW03 Birgit Pfitzmann, Michael Waidner: Federated Identity-Management Protocols — Where User Authentication Protocols May Go; 11th Cambridge Workshop on Security Protocols, Cambridge (UK), 2003, proceedings to be published by Springer, 2004
- PW03a Birgit Pfitzmann, Michael Waidner: Analysis of Liberty Single-Signon with Enabled Clients; IEEE Internet Computing 7(6) 2003, 38-44
- SAM02 Security Assertion Markup Language (SAML); OASIS Standard, November 2002, <http://www.oasis-open.org/committees/security/docs/>
- Sle01 Marc Slemko: Microsoft Passport to Trouble; Rev. 1.18, November 2001 <http://alive.znep.com/~marcs/passport/>
- WSF03 BEA, IBM, Microsoft, RSA Security, VeriSign: WS-Federation: Passive Requestor Profile; Draft, Version 1.0, July 2003, <http://www-106.ibm.com/developerworks/webservices/>